lvcreate

Créer un volume logique

lvcreate créé un nouveau volume logique dans un volume group en allouant des extents logiques dans le pool d'extents physiques disponibles dans ce volume group. S'il n'y a pas suffisamment d'extents physique, le volume group peut être étendu avec d'autres volumes physiques ou en réduisant des volumes logiques existants dans ce volume group. Si un ou plusieurs volumes physiques sont spécifiés, l'allocation d'extents physiques seront restreints à ces volumes. Le secondes forme supporte la création de volumes logiques snapshot qui conservent le contenu du volume logique original.

OPTIONS

-a|–activate [a] [e|l|s] {y|n} Contrôle la disponibilité des volumes logiques à utiliser immédiatement après que la commande soit terminées. Par défaut, les nouveaux volumes logiques sont activés (-ay). -an laisse le nouveau volume logique inactif si possible. -aay, l'autoactivation est gérée avec un match dans activation/auto_activation_volume_list dans lvm.conf. -aey active exclusivement sur un nœud et -a{all}y active seulement sur le nœud local.

–addtag Tag Ajoute le tag spécifié. Peut être spécifié plusieurs fois

–alloc AllocationPolicy Sélectionne la stratégie d'allocation (anywhere|contiguous|cling|inherit|normal)

-A|–autobackup {y|n} Backup automatiquement les métadonnées après un changement

-H|–cache Crée un volume logique cache pour pool de cache. avec –extents ou –siwe, créé un volume logique cache. quand le nom du volume group est spécifié avec un volume logique existant qui n'est pas un pool cache, un tel volume est traité comme volume cache origine et un pool cache est créé. Dans ce cas –extents ou –size est utilisé pour spécifier la taille de volume pool cache.

–cachemode {passthrough|writeback|writethrough} Spécifie un mode de cache pour les écriture dans un lv cache. writeback,

–cachepolicy Policy Seulement pour les LV cache. Définis la stratégie de cache. mq est la stratégie de base, smq est plus avancé

–cachepool CachePoolLogicalVolume Spécifie le nom du volume pool cache. L'autre manière de spécifiée le nom du pool est d'ajouter le nom du VG en argument.

–cachesettings Key=Value Seulement pour les LV cache. Définis les paramètres cache.

-c|–chunksize ChunkSize Spécifie la taille de chunk pour les LV snapshot, pool cache et pool thin. Pour les snapshots, doit être une puissance de 2 entre 4Kio (défaut) et 512Kio. Pour les pools cache la valeur doit être un multiple de 32Kio entre 32Kio et 1Gio (défaut : 64Kio). Quand la taille est spécifiée avec le volume caching, elle ne peut pas être inférieur à la taille de chunk du pool cache. Pour les pools thin la valeur doit être un multiple de 64Kio (défaut) et 1Gio. allocation/thin_pool_chunk_size_policy dans lvm.conf sélectionne une stratégie de calcul différent.

–commandprofile ProfileName Sélectionne le profile de configuration de commande à utiliser

-C|–contiguous {y|n} Définis ou redéfinis la stratégie d'alllocation contigüe pour les volumes logiques. Défaut : pas d'allocation contigüe

-d|–debug Définis le niveau de débug, de 1 à 6 fois pour augmenter les détails

–discards {ignore|nopassdown|passdown} Définie le mode d'annulation pour le pool thin. Défaut : passdown.

–errorwhenfull {y|n} Configure le comportement d'un pool thin quand l'espace est remplis. Défaut : no. Le périphérique met ses opérations en queue jusqu'au timeoute. À yes, les opération I/O sont immédiatement mis en erreur.

-l|–extents LogicalExtentsNumber [%{FREE|ORIGIN|PVS|VG} Définis la taille du volume logique en unités d'extensions logique. avec '+' la valeur est ajoutée à la taille actuelle et sans, la valeur est absolue. Le nombre total d'extents physiques alloués sera supérieur, par exemple, si le volume est mirroré. Ce nombre peut également être exprimé en % de l'espace total dans le VG avec %VG, relatif à la taille existante du LV avec %LV, de l'espace restant dans le PV avec %PVS, en pourcentage de l'espace total du volume logique d'origine avec %ORIGIN. La valeur résultante est arrondis au supérieur.

-L|–size LogicalVolumeSize Définis la taille du volume logique en unité spécifiée. Avec un '+', la valeur est ajouté, sinon elle est prise comme valeur absolue.

-i|–stripes Stripes Indique le nombre de stripes pour l'extension. Non applicable aux LV utilisant le format de métadonnée original.

-I|–stripesize StripeSize Donne le nombre de ko pour la granularité des stripes.

-K|–ignoreactivationskip Ignore le flag skip des LV durant l'activation.

–ignoremonitoring Ne tente pas d'interragir avec dmeventd sauf si –monitor est spécifié

–minor Minor [-j|–major Major] Définis le numéro majeur et mineur. Non supporté avec les volumes pools. Ignoré avec les kernels récents

–metadataprofile ProfileName Sélectionne le profile de configuration des métadonnées à utiliser

-m|–mirrors Mirrors Créé un LV mirroir avec le nombre de copies spécifiée. avec –nosync, la création du mirroir saut la resynchronisation initiale. Toutes les données après seront mirrorée, mais le contenu original ne le sera pas.

–corelog|–mirrorlog {disk|core|mirrored} –corelog est équivalent à –mirrorlog. Spécifie le type de log à utiliser pour le mode mirror. disk créé une copie persistante des données. core signifie que le mirroir est regénéré en copiant les données du premier périphérique à chaque fois que le volume logique est activé, comme après un reboot. mirrored créé un log persistant qui est lui-même mirroré

–nosync N'effectue pas la resynchro initial à la création du mirroir

-R|–regionsize MirrorLogRegionSize] Un mirroir est divisé en région de la taille spécifiée, et le log mirroir utilise cette granularité pour suivre les régions à synchroniser.

–monitor {y|n} Active le monitoring d'un mirroir, snapshot ou pool thin avec dmeventd si installé. Si un périphérique utilisé par un mirroir monitoré reporte une erreur d'E/S, l'erreur est gérée en accord avec activation/mirror_image_fault_policy et activation/mirror_log_fault_policy dans lvm.conf

-n|–name LogicalVolume Définis le nom du nouveau volume logique

–noudevsync Désactive la synchronisation udev. Le processus n'attend pas de notification udev.

-p|–permission {r|rw} Définis les permissions d'accès. Défaut : rw

-M|–persistent {y|n} À yes, le numéro mineur est persistant. Les volumes pool ne peuvent pas avoir de numéros mineur et majeur persistants.

–poolmetadatasize MetadataVolumeSize Définis la taille des métadonnées du pool. Entre 2Mio et 16Gio pour les pool thin, jusqu'à 16Gio pour un pool cache. La valeur par défaut pour un thin pool est ((Pool_LV_size / Pool_LV_chunk_size addentry articles autoadd autofind autoprod createalpha createbeta createdb createprod findentry fullpowa generator.php genhtml genman genmd gentex html insert man md pdf regen setfor setfor2 sql temp temp-new-pc tex threads ToDo 64b)

–poolmetadataspare {y|n} Contrôle la création et la maintenance de métadonnées de pool spare qui sont utilisés pour la récupération automatique du pool. Un seul volume est maintenu dans un volume group avec la taille des plus grosses métadonnées du pool.

–[raid]maxrecoveryrate Rate Définis le taux de récupération maximum pour un volume RAID. Le taux est spécifié comme quantité par seconde pour chaque périphérique dans l'array.

–[raid]minrecoveryrate Rate Définis le taux de récupération minimum pour un volume RAID. Le taux est spécifié comme quantité par seconde pour chaque périphérique dans l'array.

-r|–readahead {ReadAheadSectors|auto|none} Définis le compteur de secteur read ahead de ce volume logique.

-k|–setactivationskip {y|n} Contrôle si les volumes logiques sont flaggés pour être ignoré à l'activation. –ignoreactivationskip doit être utilisé. Le flag n'est pas appliqué durant la désactivation. Utiliser lvchange –setactivationskil pour changer le flag skip. voir activation/auto_set_activation_skip dans lvm.conf.

-s|–snapshot|-H|–cache {[VolumeGroup/]OriginalLogicalVolume Créé un LV snapshot pour un lv existant, appelé le lv d'origine. Un snapshot thin est créé quand l'origine est un volume thin et la taille n'est pas spécifiée. Un thin snapshot partage les même blocks dans le volume pool thin. Un snapshot de volume non thin avec la taille spécifié n'a pas besoin d'avoir la même quantité de stockage que l'origine, généralement 15-20% est suffisant. Note : une petite quantité d'espace allouée est utilisé pour suivre les emplacements des chunks de donnée. Si –thinpool est spécifié, un volume thin est créé et utilise le volume logique d'origine spécifié comme origine externe qui sert de blocks non-provisionnés. Seul les volumes lecture-seul peuvent être utilisés comme origine externe.

-V|–virtualsize VirtualSize Créé un périphérique provisionné léger ou un périphérique sparse. global/spase_segtype_default dans lvm.conf configure le type de segment sparse par défaut.

-t|–test Lance en mode test

-T|–thin Créé un pool thin ou un lv thin ou les 2. Avec –size et –extents, créé un lv pool thin. Avec –virtualsize, créé un lv thin dans le pool donné. Avec les 3 arguments, créé un pool thin et un lv utilisant ce pool.

–thinpool ThinPoolLogicalVolume Nom du volume pool thin.

–type SegmentType Créé un volume logique avec le type de segment spécifié. (cache, cache-pool, error, linear, mirror, raid1, raid4, raid5_la, raid5_ls (= raid5), raid5_ra, raid5_rs, raid6_nc, raid6_nr, raid6_zr (= raid6), raid10, snapshot, striped, thin, thin-pool ou zero). Défaut : linear.

-v|–verbose Mode verbeux

-W|–wipesignatures {y|n} Contrôle l'effacement des signatures détectées dans le nouveau LV. Si cette option n'est pas spécifiée, la signature est remplie de 0 avec -Z. Peut être contrôlé avec allocation/wipe_signatures_when_zeroing_new_lvs dans lvm.conf. Si l'effacement blkid est utilisé allocation/use/blkid_wiping doit être définis. Les lv lecture seul ne sont pas effacés

-Z|–zero {y|n}] Remplis de 0 les 4Kio de données du début du lv. Défaut : yes. Les lv lecture seul ne sont pas effacés.

Exemples

Créer un lv striped avec 3 stripes, une taille de stripe de 8Kio et une taille de 100Mio dans le VG vg00
lvcreate -i 3 -I 8 -L 100M vg00
Créer un lv mirroir avec 1 copie et une taille de 500Mio utilisable
lvcreate -m1 –mirrorlog core -L 500M vg00
Créer un lv snapshot vg00/snap qui a accès au contenu de vg00/lvol1 à la date de création du snapshot. Si le lv d'origine contient un système de fichier, on peut monter le snapshot dans un répertoire arbitraire pour accéder au contenu du système de fichier pour lancer une sauvegarde tout en mettant à jours le système de fichier original
lvcreate –size 100m –snapshot –name snap /dev/vg00/lvol1
Créer un lv snapshot vg00/snap avec une taille de 20% du lv d'origine
lvcreate -s -l 20%ORIGIN –name snap vg00/lvol1
Créer un périphérique sparse nommé /dev/vg1/sparse de 1Tio avec un espace de 100Mio de données actuelles
lvcreate –virtualsize 1T –size 100M –snapshot –name sparse vg1
Créer un lv linéaire vg00/lvol1 utilisant les extents physiques /dev/sda :0-7 et /dev/sdb :0-7 pour l'allocation des extents
lvcreate -L 64M -n lvol1 vg00 /dev/sda :0-7 /dev/sdb :0-7
Créer un lv Raid5 5Gio vg00/my_lv avec 3 stripes (plus un disque de parité) et une taille de stripe de 64Kio
lvcreate –type raid5 -L 5G -i 3 -I 64 -n my_lv vg00
Créer un lv RAID5 vg00/my_lv, utilisant tout l'espace disque dans le VG et répartis sur tous les PV dans le VG
lvcreate –type raid5 -l 100%FREE -n my_lv vg00
Créer un RAID10 de 5Gio vg00/my_lv avec 2 stripes dans 2 mirroirs. Noter que les arguments -i et -m fonctionnent différemment :
lvcreate –type raid10 -L 5G -i 2 -m 1 -n my_lv vg00
Créé un lv pool de 100Mio pour du provisionning léger avec 2 stripes de 64Kio et un taille de chunk de 2156Kio avec un volume de 1Tio
lvcreate -i 2 -I 64 -c 256 -L100M -T vg00/pool -V 1T –name thin_lv
Créé un lv snapshot thin "thinsnap" du volume thin "thinvol" qui partagent les même blocks dans le pool thin. Noter que la taille ne doit pas être spécfiée sinon un snapshot non-thin est créé :
lvcreate -s vg00/thinvol –name thinsnap
Créer un volume snapshot thin d'un volume lecture seule inactif "origin" qui devient ensuite l'origine externe pour ce snapshot
lvcreate -s –thinpool vg00/pool origin
Créer un lv pool cache qui peut être utilisé pour cacher un volume logique
lvcreate –type cache-pool -L 1G -n my_lv_cachepool vg /dev/fast1
S'il y a un lv pool cache exsistant, créer un grand périphérique lent (le lv origine) et le lier au lv pool cache, créant un lv cache
lvcreate –cache -L 100G -n my_lv vg/my_lv_cachepool /dev/slow1
S'il y a un lv existant, créer le lv pool cache et le lier avec ce lv, créant un lv cache
lvcreate –type cache -L 1G -n my_lv_cachepool vg/my_lv /dev/fast1